Inserting a Fast Walsh–Hadamard Transform Between ReLU Layers

(via the implicit decision matrix view)

Consider a standard ReLU layer written in the “implicit decision matrix” form:

$y = D(x)Wx$

- W: weight matrix
- D(x): diagonal “decision matrix” induced by ReLU
    - entries are 0 or 1 depending on activation signs
- Interpretation: D(x) **selects a subset of rows** of (W), i.e. it gates which features survive

---

Standard behavior (no transform)

In a stack of ReLU layers:

$y = D_2W_2D_1W_1x$

Each D_i **zeros out coordinates in the intermediate representation**, which is equivalent to:

- removing columns of the *next* weight matrix
- so parts of W_2 are **never used for that input**

👉 This is the key inefficiency:
ReLU gating sparsifies the *input* to the next layer, so only a subset of parameters participate.

Insert a Walsh–Hadamard transform

Now insert a Fast Walsh–Hadamard Transform (FWHT) between layers:

$y = D_2W_2HD_1W_1x$

where:

- H is the Hadamard matrix (implemented with O(nlogn) FWHT)
- optionally preceded by random sign flips to avoid spectral bias

---

Reinterpreting the decision matrix

Focus on the middle part:

HD_1

In the standard basis:

- D_1 zeros coordinates → removes columns of W_2

But in the Hadamard basis:

- D_1 acts **before mixing**
- so after multiplication by H, the effect is:

  columns of H are zeroed instead of columns of W_2

Key shift in viewpoint

Without H:

- D_1 → zeros coordinates
- → zeros **columns of W_2**
- → many parameters in W_2 are unused

With H:

- D_1 → zeros coordinates
- → equivalent to zeroing **columns of H**
- H mixes everything through weighted combinations of the remaining transform columns

- → W_2 now sees a **dense input**

👉 So:

> **The gating is applied in the transform domain, not in parameter space.**

Consequence: all parameters in W are used

Because H spreads each surviving coordinate across *all* output dimensions:

- Every neuron in the next layer receives contributions
- Every column of W_2 is engaged
- No parameter is systematically bypassed

In contrast to standard ReLU:

- where dead activations → dead columns → unused weights

Intuition

Think of it like this:

- ReLU produces a **mask** (the decision matrix)
- Normally that mask acts in the **coordinate basis**
- With FWHT, the mask acts in a **global, distributed basis**

So instead of:

- “turning off features”

you get:

- “turning off global patterns (Walsh modes)”

Effect on information flow

This has several important implications:

1. Improved parameter utilization

- Every layer uses its full parameter capacity for every input
- Reduces “conditional underuse” of weights

2. Better information mixing

- FWHT is maximally mixing (like a cheap FFT)
- Prevents locality of information loss due to ReLU gating

3. Reduced fragmentation of representation

- Standard ReLU networks:
  - carve space into regions
  - but also isolate subspaces
- With FWHT:
  - representations remain globally entangled

4. Shift in what gating means

- Standard:
  - gating = selecting coordinates
- With FWHT:
  - gating = selecting **spectral components / global patterns**

Interpretation in associative memory terms

Using your hierarchical associative memory framing:

- Standard ReLU:
  - retrieves partial patterns
  - but suppresses unused channels → information loss
- With FWHT:
  - retrieved patterns are **redistributed globally**
  - so downstream layers can still access all recalled information

👉 The network behaves less like:

- a sequence of lossy filters

and more like:

- a **conditionally modulated global memory system**

Potential effects in practice

Likely benefits

- Better gradient flow (fewer dead pathways)
- Higher effective capacity per parameter
- More uniform learning across weights
- Reduced reliance on dense skip connections for information flow

Possible downsides

- Loss of locality / interpretability
- Harder specialization of neurons
- Risk of over-mixing (noise propagation)
- May require normalization or structured randomness (e.g. sign flips)

One-line summary

> Inserting a Walsh–Hadamard transform shifts ReLU gating from *parameter space* (zeroing columns of (W)) to *representation space* (zeroing modes of (H)), ensuring that **all parameters are used for every input** and turning the network into a more globally coupled, conditionally modulated system.

First few FWHT equivalent matrices:

$$H_1 = [1]; \qquad H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix};$$

$$H_4 = H_2 \otimes H_2 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$H_8 = H_2 \otimes H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$